Journal of Hydrosciences and Environment

ISSN: 2345-5608

# A Combined Adaptive Differential Evolution Algorithm with Ensemble of Mutation Strategies for Reducing Irrigation Deficiencies

Hossein Mohammadi*[1]

1.*Master of Science, Department of Computer Science, Faculty of Mathematics - Statistics and Computer Science, University of Sistan and Baluchestan, Zahedan, Iran.

**ARTICLEINFO**

**Article history:**

**Received:** 27 April 2022

**Accepted:** 3 May 2023

**Keywords:**

*Water Resource Management,*

*Differential Evolution,*

*Ensemble of Mutation Strategies,*

*Adaptive and Self-adaptive Parameters,*

*Single-reservoir Allocation Optimization*

**ABSTRACT**

The optimal utilization of dams water resources in order to meet the water needs of different departments is an important problem in the management and engineering of water resources. Evolutionary algorithms have provided great success to create balancing in supply of water resources shortages with the goal of controlling the water resources storage and dams release rates. Differential evolution (DE) is an efficient search technique and method to solve numerical optimization problems and real-world applications. DE suffers from numerous weaknesses that usually due to the use of a mutation strategy and consideration of constant settings for parameters. In this paper, an adaptive DE algorithm titled Combined Improved Multi-Population Ensemble DE (CIMPEDE) is suggested. In CIMPEDE, the entire population is divided into four sub-populations; three mutation strategies are considered to achieve a better, and two adaptive and self-adaptive schemas are proposed for setting parameters. CIMPEDE has been able to achieve the optimal solution with the highest quality by choosing the best mutation strategy in each repetition, dynamic adjustment of parameters and exchange of information between sub-populations. For prove the performance of CIMPEDE, extensive experiments have been performed on mathematical benchmark functions (entitled CEC 2013) and a real-world problem for modeling of the single-reservoir system of Golestan dam in Iran with the aim of decreasing irrigation deficiencies. CIMPEDE has also made comprehensive comparisons with DE-based applied approaches. In the case of Golestan dam problem, mean absolute error between real demand and released water by CIMPEDE was zero MCM. While, the comparable algorithm in second rank could not get a value better than 1.81E-220 MCM. Also, in test of reliability and vulnerability indices and test of mean violations intensity between demand and released water in an annual average, the best results were recorded among the comparable evolutionary algorithms for CIMPEDE.

## 1. Introduction

Today, with the development of science and technology, various scientific fields have faced with various optimization problems. Each optimization problem is defined with a number of decision-making variables. For each decision-making variable, a range of real values is considered to constitute the search space of the problem. The objective of optimization is to choose the best solution among a set of solutions in the search space (Bazaraa et al., 2008). In general, there are several methods, such as evolutionary algorithms, for solving optimization problems. Evolutionary algorithms are usually population-based approaches (a set of solutions). These algorithms evolve the population members repeatedly. Thus, in each repetition of the algorithm process, the position of the population members in the search space is changed and, by identifying the desired area, efficiency increases among members of the population (Boussaïd et al., 2013). Some of the population-based evolutionary algorithms include: Ant Colony Optimization (ACO) (Dorigo & Di Caro, 1999), Genetic Algorithm (GA) (Holland, 1992), Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995), and Differential Evolution (DE) (Storn, 1996; Storn & Price, 1997).

DE is one of the most efficient, easiest and effective evolutionary algorithms (Ali et al., 2015; Storn & Price, 1997), and is a population-based random search method. DE consists of two parts of an initial population and evolutionary loop. This algorithm improves the population to a global optimization in an evolutionary loop using three mutation, crossover, and selection operators (Xiang et al., 2015). DE is very similar to classical genetic algorithm (GA). In GA, each solution is known as a chromosome. Each decision-making variable of the optimization problem is also a gene on the chromosome. GA randomly generates an initial population of chromosomes and uses three selection, crossover, and mutation operations in an evolutionary loop. In the selection operation based on a selection strategy, two parent solutions are selected. Then, in the crossover operation, based on a single-point cutting strategy from two parent solutions selected, two child solutions aregenerated. In the mutation operations, in order to increase diversity, a number of solutions are randomly selected from the offspring population, and some genes of them are randomly reset. Finally, from the parent and offspring populations, the best solutions are passed on to the next generation as large as the initial population. The evolution loop is repeated until the termination condition is observed (Holland, 1992). The classical DE approach is described in detail in Section 2.3. By studying section 2.3, we will clearly see the difference between the DE and the GA presented in this paragraph.

Classical DE (Storn, 1996) has a very poor performance in dealing with many complex problems with a large number of decision-making variables. In general, high accuracy, reduced computational time and convergence into a solution sufficiently close to the optimal global solution are the main objectives of solving optimization problems. Therefore, the combination of DE with many efficient techniques can provide smarter solutions throughout the evolutionary process.

The population size (NP) is an important factor in improving the performance of DE. As a large population can create a greater diversity among the population members and a better exploration of the search space for the algorithm (Salehinejad et al., 2017). Also, DE's performance is mainly dependent on the strategies of mutation, crossover, selection and their relative parameters' settings, including mutation scale factor (F) and crossover rate (CR) (Wu et al., 2016).

Different mutation strategies for DE are presented in (Cui et al., 2016; Storn & Price, 1997). In DE, the proper selection of mutation strategies is different for solving different optimization problems. In fact, each mutation strategy has distinct features that provide different behaviors for each problem. It has been proven experimentally that using DE/curr2best/1 mutation strategy results in good convergence rate and excellent performance to solve unimodal problems. However, this strategy does not converge to a high-quality solution to solve multi-modal problems. In such a situation and in the face of multi-modal problems, the DE/rand/1 mutation strategy seems to be more appropriate (Wu et al., 2016). This strategy, with a slow rate of convergence, however, has a strong exploration capability that prevents premature convergence and stagnation phenomena (Qin et al., 2008). DE/curr2rand/1 mutation strategy, which is usually implemented regardless of crossover operation in DE body, is an appropriate strategy for solving problems that their objective functions have been rotated (Cui et al., 2016). The parameter settings can

*Corresponding author's
email:Mohammadi_cs94@pgs.usb.ac.ir

also have a great effect on the performance of DE. So that, achieving the optimal solution in solving any problem using DE can be achieved by specially adjusting of the parameters in DE. Even, it is possible that, Finding the optimal answer in solving a special problem using DE during the evolutionary loop and in the moment be require different parameter settings (Qin et al., 2008). In fact, small CR values are suitable for solving problems with a separable feature, and large CR values have been shown to have a significant effect on solving problems with a non-separable feature. Also, in order to control the population diversity, at the early stages of the evolutionary process, large values of F have been proposed and, in order to accelerate the population convergence, at the final stages of the evolutionary process, small values of F have been proposed (Cui et al., 2016; Wu et al., 2016).

Today, several techniques have been proposed to increase DE performance. The techniques include defining new mutation strategies and improving existing mutation strategies (Cheng et al., 2021; Deng et al., 2021; Li et al., 2020; Zhou et al., 2022), employing an ensemble of mutation strategies based multi-population structure (Ali et al., 2015; Cui et al., 2016; Li et al., 2021; Ma & Bai, 2020; Wu et al., 2016; Xu et al., 2019), and presenting adaptive and self-adaptive schemes for parameter settings (Draa et al., 2015; Duan et al., 2019; Fan et al., 2019; Mallipeddi et al., 2011; Song & Li, 2021; Wang et al., 2011; Zhang & Sanderson, 2009).

This paper proposes an algorithm titled "Combined Improved Multi-Population Ensemble Differential Evolution" (CIMPEDE) in order to control the proper trade-off between diversity and convergence among the population members and increase performance of DE in solving various optimization problems with real-world applications. CIMPEDE approach uses multi-population structures, ensemble of mutation strategies, and adaptive and self-adaptive schema of the parameters. The proposed approach is an improved version of MPEDE algorithm, which is extensively expressed in (Wu et al., 2016). Also, in CIMPEDE body, SinDE algorithm is used, with more details in (Draa et al., 2015).

The performance of the proposed algorithm is examined by CEC 2013 benchmark function suite on dimensions 30 and 50, and its performance is compared with some of the improved DE versions. In addition, in order to evaluate the quality of the proposed algorithm performance in solving real-world problems, problem of optimizing the allocation of a single-reservoir system in Golestan Dam is intended to minimize deficiencies.

The organization of the following sections is as follows: In Section 2, the basic concepts and classical DE details are studied. In Section 3, the proposed approach CIMPEDE is introduced in great detail. In Section 4, the discussion of experimental results is analyzed. Finally, in Section 5, the conclusion is presented.

## 2. Materials and Methods

### 2.1. Mathematical model of an optimization problem

The mathematical model of an unconstrained optimization problem and a constrained optimization problem are presented as Eq. (1) and Eq. (2) (Zhang et al., 2017):

$$\begin{aligned} &Minimize\ (Maximize) f(X) \\ &where: \\ &\quad X = (x_1, x_2, \dots, x_j, \dots, x_D) \in S \\ &\quad S \subseteq R^n \\ &\quad j = 1, 2, \dots, D \end{aligned} \tag{1}$$

So that the optimization problem is defined in n-dimensional Euclidean space titled $S$ search space. $R^n$ is a n-dimensional search space in a set of real numbers. Each $X$ is a solution in search space and vector with $D$ variable as the decision-making variable. Also, the value evaluated by the objective function for the solution $X$ in the search space is determined by $f(X)$.

$$\begin{aligned} &Minimize\ (Maximize) f(X) \\ &where: \\ &\quad X = (x_1, x_2, \dots, x_j, \dots, x_D) \in S \\ &\quad S = [X | G_i(X) \leq 0] \\ &\quad i = 1, 2, \dots, M \\ &\quad j = 1, 2, \dots, D \end{aligned} \tag{2}$$

So that $G_i(X)$ are functions of problem constraints, $M$ denotes the number of constraint functions.

The objective of optimization is also to find a solution such as $X^*$ that $X^* \in S$ and satisfy the condition $f(X^*) \leq f(X)\ \forall X \in S$ for a minimization problem.

### 2.2. The optimization of water reservoir allocation in the Golestan Dam

Surface water reservoirs seem to be very important in different sectors of agriculture, drinking and

industry. The problem of optimal utilization of water reservoirs, due to the time of river discharge and its mismatch with the existing requirements of different applicant sectors, can cause damage to different applicant sectors. Hence, the optimization of utilization of water reservoirs is one of the most important problems of engineering, agriculture and economics. On the other hand, optimization of allocation of water reservoirs has high complexity usually requiring high costs to solve using definitive mathematical optimization methods. For this purpose, many researchers have been using evolutionary algorithms, neural networks, and etc. to solve the optimal utilization problem of water reservoirs (Ehteram et al., 2018; Ruigar & Golian, 2016; Saadat et al., 2011; Sharifi et al., 2012).

Golestan Reservoir Dam is located on Gorganroud River, in Golestan Province, northern Iran. Gorganroud River extends 359 km into the Caspian Sea. Fig. 1 shows the position of this river in the area (Ehteram et al., 2018). The reservoir dam had a capacity of 86 MCM at the beginning of the construction and the reservoir volume was 135 MCM when flooding. Today, the capacity of Golestan Dam due to the volume of sediments deposited has been reached 62 MCM (Ehteram et al., 2018).

The main features of Golestan Dam, including inlet flows, downstream requirements, water evaporation, and etc., are discussed in detail in (Ehteram et al., 2018). In the remainder of the whole, the above problem is expressed as follows.

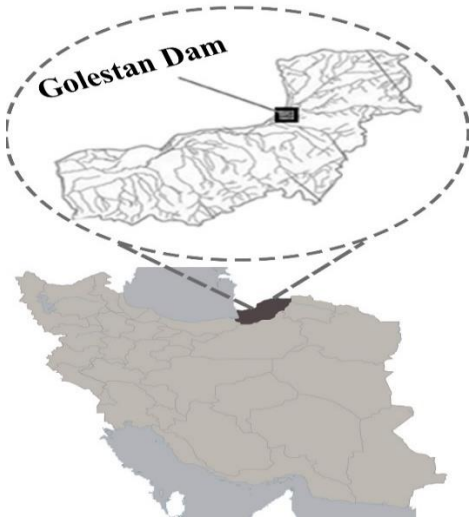



**Fig. 1.** Golestan Reservoir Dam position on Gorganroud River (Ehteram et al., 2018)

The objective function of the reservoir dam optimization problem ($OF$) by assuming minimization of deficiencies and / or maximizing the amount of supply needs of different applicant sectors, is defined in accordance with Eq. (3):

$$OF = \sum_{t=1}^{T} \left(\frac{D_i - R_i}{\max(D_i)}\right)^2 \quad (3)$$

Where $\max(D_i)$ is maximum downstream demand of reservoir in the total operation periods. $D_i$ is downstream requirements and $R_i$ is the reservoir release value. $t = 1, 2, \dots, T = 120$ also shows consecutive periods; so that for each period, the recorded data are according to different parameters at the end of each month of the year. The data are about the past ten years, which have been collected from Regional Water Organization of Golestan Province.

The equation of continuity of the water system in the reservoir of Golestan Dam is also expressed as Eq. (4):

$$S_{t+1} = S_t + I_t - E_t - R_t - Sp_t \quad (4)$$

Where $S_{t+1}$ is the reservoir storage at the end of period $t$ (in terms of MCM) and $S_t$ is the storage of the beginning of each period $t$ (in terms of MCM). The parameters $I_t$, $E_t$, $R_t$ and $Sp_t$, respectively, are the inlet flow into the reservoir, evaporation from the reservoir surface, reservoir release and overflow from the reservoir surface during period $t$ (in terms of MCM).

The evaporation volume in each period is a function of the reservoir surface and evaporation height. The reservoir surface is calculated in terms of surface-volume-height data of Golestan Dam. The release of the reservoir ($R$) in each period $t$ is also some in the interval of Eq. (5).

$$R_{min} \leq R_t \leq R_{max} \quad (5)$$

Where $R_{min}$ is lower limit of release and is set to 0, and $R_{max}$ is the upper limit of the release, which is equal to the maximum volume of the reservoir.

The volume of the reservoir is also defined as Eq. (6):

$$S_{min} \leq S_t \leq S_{max} \quad (6)$$

Where $S_{min}$ is the minimum volume of the reservoir and $S_{max}$ represents the maximum volume of the reservoir.

The parameters of the reservoir inlet flow, reservoir release, reservoir evaporation and overflow from the reservoir surface can violate the reservoir's water

volume in Eq. (6). Therefore, when the volume of the reservoir ($S_t$) is unauthorized in each period $t$, it is penalize using Eq. (7):

$$\begin{cases} if\ S_t \prec S_{min} \quad then \\ P(1) = K \times \sum_{t=1}^{T} \left( {(S_{min} - S_t)^2} \Big/ {S_{min}} \right) \\ if\ S_t \succ S_{max} \quad then \\ P(2) = K \times \sum_{t=1}^{T} \left( {(S_t - S_{max})^2} \Big/ {S_{max}} \right) \end{cases} \quad (7)$$

Where $K$ is a large scalar, and $P(1)$ and $P(2)$ compute the rate of penalty functions.

### 2.3. Classic differential evolution algorithm

DE algorithm developed by Storn and Price (1995) has a structure consisting of an initial population and evolutionary loop (three mutation, crossover and selection operators) (Storn, 1996; Storn & Price, 1997). Also, the three parameters of mutation scale factor (F), crossover rate (CR) and population size (NP) are the basic parameters of DE. Further, DE will be examined in more detail.

For create the initial population generation we have that: Assuming that each member of the population of generation $t$ represents a solution in the search space of a problem with $D$ as a decision-making variable, it is represented by $X_i^t = (x_{1i}^t, x_{2i}^t, \dots, x_{Di}^t)$ vector for $i = 1, 2, \dots, NP$. In this case, the initial population or zero generation is represented by $X_i^0 = (x_{1i}^0, x_{2i}^0, \dots, x_{Di}^0)$ symbol and is usually generated randomly based on the normal statistical distribution. So that $x_{lb_j} \le x_{ji}^0 \le x_{ub_j}$ for $i = 1, 2, \dots, NP$ and $j = 1, 2, \dots, D$, and each decision-making variable $x_{ji}^0$ is determined by Eq. (8) (Ali et al., 2015; Storn & Price, 1997):

$$x_{ji}^0 = rand(0,1) \times \left( x_{ub_j} - x_{lb_j} \right) + x_{lb_j} \quad (8)$$

Where $rand(0,1)$ is a random number with a uniform distribution at the interval of $[0, 1]$ and $x_{lb_j}$ and $x_{ub_j}$ respectively, are lower and upper limits of decision-making variable $j$.

In the mutation operation we have that: Assuming that the current generation is the generation $t$, and the population of this generation is displayed with $X_i^t = (x_{1i}^t, x_{2i}^t, \dots, x_{Di}^t)$ for $i = 1, 2, \dots, NP$. The mutation operation creates the mutant vector $V_i^t$ for $i = 1, 2, \dots, NP$ based on the population of the current generation. The vector $V_i^t$ is based on the different strategies of mutation, by random selection of distinct members from the population of generation $t$ and their combination with each other (Storn & Price, 1997; Xiang et al., 2015). The process of generating mutated vector using the three mutation strategies that are commonly used in studies (Cui et al., 2016; Storn & Price, 1997), is shown in Eqs. (9) to (11):

$$V_i^t = X_{r1}^t + F \times (X_{r2}^t - X_{r3}^t) \quad (9)$$

$$V_i^t = X_i^t + F \times (X_{best}^t - X_i^t + X_{r1}^t - X_{r2}^t) \quad (10)$$

$$V_i^t = X_i^t + K \times (X_{r1}^t - X_i^t) + F \times (X_{r2}^t - X_{r3}^t) \quad (11)$$

Where $X_{r1}^t$, $X_{r2}^t$ and $X_{r3}^t$ are three distinct members and differ from $X_i^t$ of members of the population of generation $t$, $X_{best}^t$ is the best members of the population of generations $t$, and $F$ and $K$ are real values at the interval of $[0, 1]$.

In the crossover operation we have that: The crossover operation is presented in order to introduce some vector components of the current member of population ($X_i^t$) in the mutant vector $V_i^t$. This operation produces from the combination of two vectors $V_i^t$ and $X_i^t$ the vector $U_i^t = (u_{1i}^t, u_{2i}^t, \dots, u_{Di}^t)$ in accordance with Eq. (12) (Storn & Price, 1997; Xiang et al., 2015):

$$u_{ji}^t = \begin{cases} v_{ji}^t & if\ rand_j(0,1) \le CR\ \ OR\ \ j = j_{rand} \\ x_{ji}^t & otherwise \end{cases} \quad (12)$$

Where $j = 1, 2, \dots, D$, $j_{rand}$ a random number of the set [1, 2, ..., D] and $CR$ is a real value at the interval of $[0, 1]$.

In the selection operation we have that: Selecting members to move to the next generation is usually based on the value of their objective function. Assuming that the given problem is a minimization problem, selection operation victoriously with the competition between the population of generation $t$ and corresponding solutions produced by $U^t$, forms population of generation $t + 1$ in accordance with Eq. (13) (Storn & Price, 1997; Xiang et al., 2015):

$$X_i^{t+1} = \begin{cases} U_i^t & if\ f(U_i^t) < f(X_i^t) \\ X_i^t & otherwise \end{cases} \quad (13)$$

Where $f(U_i^t)$ and $f(X_i^t)$ respectively, are the value of the objective function evaluation according to the vectors $U_i^t$, $X_i^t$ and $X_i^{t+1}$ of member $i$ of the

population of generation $t+1$.

### 2.4. Methodology

In this section, we summarize two algorithms MPEDE (Wu et al., 2016) and SinDE (Draa et al., 2015). Then, two techniques of multi-population structures (Ali et al., 2015) and adaptive scheme of parameters (Cui et al., 2016) are expressed. Finally, the proposed approach CIMPEDE is described and its pseudocode is shown in Algorithm (1).

#### 2.4.1. MPEDE algorithm

MPEDE algorithm for increasing DE performance considers an ensemble of mutation strategies and adaptive scheme of parameters. In this approach, first, in a randomized and uniform distribution, the entire population is divided into two "indicator" (with a moderate population size) and a "reward" (with a large population size) sub-populations. Each sub-population based on the mutation strategy introduced in Eqs. (9) to (11) and adaptive scheme of the parameters, with $nt$ repetition, addresses the evolution of its members. Then, reward sub-population chooses the mutation strategy that has the best performance during $nt$ repetition. In other words, in MPEDE, during the evolutionary process, the best mutation strategy gains more computational resources and makes it possible to achieve faster access to a global optimal solution. In MPEDE, an ensemble of mutation strategies is selected for trade-off between diversity and convergence among the population members and better exploration in search space.

The formation of sub-populations in MPEDE is based on each repetition of the evolutionary process of the algorithm. Each member of the population is randomly assigned to a sub-population including $pop_1, pop_2, and\ pop_3$. Also, at the end of each repetition of the evolutionary process, members of the sub-populations are combined together. These members are divided into three sub-populations of $pop_1$, $pop_2$ and $pop_3$ based on a uniform random distribution, for generation t+1.

#### 2.4.2.SinDE algorithm

SinDE algorithm is presented for increasing classical DE performance. The mutation and crossover strategies in SinDE are updated in accordance with Eqs. (9) and (12), respectively. In addition, setting parameters in SinDE is implemented using a self-adaptive scheme based on a sinusoidal model. The sinusoidal model, using a sinusoidal formula, modifies the values of the parameters periodically (increasing or reducing) at the interval of $[0, 1]$. The setting of the parameters based on the sinusoidal model is defined according to Eq. (14) and its variation graph is shown in Fig. 2.

$$\begin{cases} F = \frac{1}{2}\left(\sin(2\pi t \times freq)\frac{t}{Max_t} + 1\right) \\ CR = \frac{1}{2}\left(\sin(2\pi t \times freq + \pi)\frac{t}{Max_t} + 1\right) \end{cases} \quad (14)$$

Where $freq$ represents the frequency of the sinusoidal function, $Max_t$ is the maximum number of repetitions of the algorithm and $t$ is the current repetition of the algorithm.

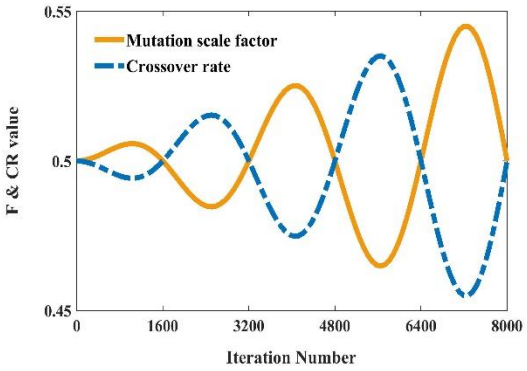



**Fig. 2.** Periodic change of parameters F and CR based on Sinusoidal model

#### 2.4.3.The technique of multi-population structures (Ali et al., 2015)

In (Ali et al., 2015), multi-population structures are based on the two concepts of "isolation stage" and "migration stage". At the isolation stage, the entire population is randomly divided into several sub-populations. At the migration stage, also based on methods (such as random and roulette wheel selection), some members of each sub-population are selected and transferred to other sub-populations based on pre-designed topologies. In one of these methods, some members of each sub-population are identified based on random selection and transferred to another sub-population according to complete network topology. Fig. 3 shows a complete network topology with four sub-populations. In complete

network topology assuming n sub-populations, each selected member can be transferred to any of n-1 sub-population (Ali et al., 2015).

**2.4.4.The adaptive scheme technique (Cui et al., 2016)**

Using the adaptive scheme of parameters to set two parameters $F$ and $CR$ in (Cui et al., 2016) is as follows: In the generation $t$ of the evolutionary process, $F_i^t$ for member $i$ is determined using Cauchy distribution in accordance with Eq. (15):

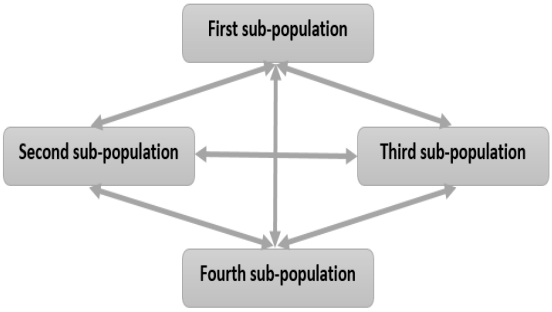



**Fig. 3.** An example of complete network topology

$$F_i^t = Cauchy(F_m^t, 0.1), \quad i = 1, 2, \dots, NP \qquad (15)$$

Where $Cauchy(F_m^t, 0.1)$ is a real number generated at the interval of $[0, 1]$ based on Cauchy distribution with mean $F_m^t$ and standard deviation of 0.1. In Eq. (15), if $F_i^t > 1$, the parameter $F_i^t$ is reduced to 1, and if $F_i^t < 0$, $F_i^t$ is re-generated. Also, the values of mutation scale factors associated with newly generated solutions that have succeeded in population of generation $t + 1$ are stored in $S_F$ set. The value of the parameter $F_m^t$ is initially equal to the constant 0.5 and is updated at the end of each repetition, according to Eq. (16) (Cui et al., 2016):

$$F_m^{t+1} = \begin{cases} (1 - w_F).F_m^t + w_F.mean_L(S_F) & if\ S_F \neq \emptyset \\ F_m^t + \frac{1}{2}.\tan\left(\pi.rand(0,1) - \frac{1}{2}\right) & if\ S_F = \emptyset \end{cases} \qquad (16)$$

Where $w_F$ is the random weight factor at the interval of $[0, 1]$ and $mean_L(.)$ is the mean of Lehmer (Cui et al., 2016; Islam et al., 2011; Zhang & Sanderson, 2009) with the formula

$$mean_L(S_F) = \frac{\sum_{F_i^t \in S_F} {F_i^t}^2}{\sum_{F_i^t \in S_F} F_i^t}.$$

In the generation $t$ of the evolutionary process, $CR_i^t$ for the member $i$ of the population is initialized using Gaussian distribution according to Eq. (17):

$$CR_i^t = Gaussian(CR_m^t, 0.1), \quad i = 1, 2, \dots, NP \qquad (17)$$

Where $Gaussian(CR_m^t, 0.1)$ produces a real number at the interval of $[0, 1]$, based on Gaussian distribution with mean $CR_m^t$ and a standard deviation of 0.1. In Eq. (17), if $CR_i^t < 0$, $CR_m^t$ is set to 0, and if $CR_i^t > 1$, it will be reduced to 1. Also, the values of crossover rates associated with newly generated solutions that are located in the population of generation $t + 1$ are stored in $S_{CR}$ set. The parameter $CR_m^t$ is initially initialized with constant 0.5 and updated at the end of each repetition, according to Eq. (18):

$$CR_m^{t+1} = \begin{cases} (1 - w_{CR}).CR_m^t + w_{CR}.mean_L(S_{CR}) & if S_{CR} \neq \emptyset \\ CR_m^t + 0.1.randn & if S_{CR} = \emptyset \end{cases} \qquad (18)$$

Where $randn$ is a random number using the standard normal distribution and $w_{CR}$ is the random weight factor at the interval of $[0, 1]$.

**2.4.5.The proposed algorithm CIMPEDE**

In order to improve DE performance and achieve a high-quality solution, CIMPEDE approach is introduced using a multi-population structure, ensemble of efficient mutation strategies, and adaptive and self-adaptive schemes of the parameters. CIMPEDE approach in its body uses SinDE algorithm, multi-population structure technique described in (Ali et al., 2015), and adaptive schemes technique of parameters in accordance with (Cui et al., 2016) and provides an improved version of MPEDE. In CIMPEDE implementation process, after generating the initial population and before entering the evolutionary loop of the algorithm, the entire population is randomly divided into two "indicator" sub-population (medium-sized population) and a "reward" sub-population (large-sized population). In the other words based on Eq. (19):

$$\begin{aligned} &pop = \sum_{k=1}^{3} pop_k \\ &np_k = \lambda_k \times NP, \quad k = 1,2,3 \\ &\sum_{k=1}^{3} \lambda_k = 1 \\ &\lambda_1 = \lambda_2 = 0.2\,, \lambda_3 = 0.6 \end{aligned} \tag{19}$$

Where, $pop$ is the entire population, $NP$ is the population size of $pop$, $pop_k$ is sub-population $k$ and $np_k$ is $pop_k$ sub-population size. Finally, $\lambda_k$ is numerical constant in range $(0,1)$ for the sub-population of $k$.After the formation of sub-populations, in the evolutionary loop of the algorithm each of the proposed mutation strategies in Eqs. (9) to (11) is assigned randomly to each sub-population. The Eq. (11) mutation strategy is implemented, regardless of crossover operation. The Eq. (10) mutation strategy also uses an archive of the best solutions. So that the Eq. (10) mutation strategy chooses the participant's solutions to provide a mutant vector from the population of the current generation and archive population (Wu et al., 2016). In the following and every repetition of the evolutionary loop, setting the parameters of $F$ and $CR$ for each member of the sub-populations is based on the adaptive scheme using two Cauchy and Gaussian distributions. After the evolution of the members of each sub-population, the continuation of CIMPEDE implementation process in the same current repetition is as follows. With selecting .02 the best members of each sub-population, members of "elitist" sub-population are formed. The elitist sub-population generates new solutions by applying the Eq. (9) mutation strategy and self-adaptive scheme of $F$ and $CR$ parameters based on sinusoidal model that are suitable for improving convergence and focusing on the desired area of search space.

Algorithm 1. Pseudo code of CIMPEDE

Output: Best solution

Set $F_{m_k}^0 = 0.5$, $CR_{m_k}^0 = 0.5$, $\Delta f_k = 0$ and $\Delta Fes_k = 0$ for each $k = 1,2,3$; $freq = 0.25$;
Initialize, $NP, nt, np_{elitist_k}$ for each $k = 1,2,3$;
Initialize the pop randomly distributed in the search space;
Initialize $\lambda_k$ (randomly) and set $np_k = \lambda_k . NP$;
Divide the pop randomly into $pop_1$, $pop_2$, and $pop_3$ with respect to their sizes;
Set $t = 0$;
**while** $t \leq Max_t$
    $t = t + 1$;
    **for** k = 1 → 3
        Calculate $F_{m_k}^t$ and $CR_{m_k}^t$;
        Calculate $F_{i_k}^t$ and $CR_{i_k}^t$ for each individual $X_i^t$ in $pop_k$;
        Perform the $k$th mutation strategy and related crossover operators over sub-population $pop_k$;
        Set $S_{F_k} = \emptyset$ and $S_{CR_k} = \emptyset$;
    ***end***
    **for** i = 1 → NP
        **if** $f(X_i^t) \leq f(U_i^t)$
            $X_i^{G+1} = X_i^G$;
        **else**
            $X_i^{t+1} = U_i^t$; $\Delta f_k = \Delta f_k + f(X_i^t) - f(U_i^t)$;
            $F_{i_k}^t \rightarrow S_{F_k}$; $CR_{i_k}^t \rightarrow S_{CR_k}$;
        ***end***
    ***end***
    Sorting, and selecting the best individuals of each sub-population with size $np_{elitist_k}$ for each $k = 1,2,3$;
    Remove the best selected individuals from any sub-population;
    Consider the $I_{sp}$ index for each selected individual ($I_{sp}$ shows the sub-population that associated with each member);
    Combine the best selected individuals and form $pop_{elitist}$ with a population size of $np_{elitist}$;
    $F^t = \frac{1}{2} \times \left(sin(2\pi \times freq \times t) \times \frac{Max_t - t}{t} + 1\right)$;
    $CR^t = \frac{1}{2} \times \left(sin(2\pi \times freq \times t + \pi) \times \frac{Max_t - t}{t} + 1\right)$;
    Perform the DE/rand/1 mutation strategy and binomial crossover operators over sub-population $pop_{elitist}$;
    **for** i = 1 → $np_{elitist}$
        **if** $f(X_i^t) \leq f(U_i^t)$

```
        | | | X_i^{t+1} = X_i^t;
        | | else
        | | | X_i^{t+1} = U_i^t;
        | | end
        | end
        | Transferring any individual of sub-population pop_elitist to sub-populations pop_k using the I_sp index;
        | if mod(t, nt) == 0
        | | K = arg( max_{1≤k≤3} (Δf_k / ΔFes_k) );
        | | λ_{j=1,2,3} = 0.2;  λ_K = λ_K + 0.4;
        | | Δf_k = 0;
        | end
        | Migration individuals of sub-populations using complete net topology;
    end
```

Finally, at the end of each repetition of the evolutionary process, some members of each sub-population are transferred to other sub-populations by random selection and complete network topology.

Also, after each $nt$ repetition during the evolutionary process, the best mutation strategy is identified to be assigned to the "reward" sub-population for the next $nt$ repetition. In this situation, each "indicator sub-population" randomly selects one of the remaining mutation strategies for the next $nt$ repetition. In order to determine the best mutation strategy, Eq. (20) is used (Wu et al., 2016):

$$\frac{\Delta f_k}{\triangle Fes_k},\ \mathrm{k} = 1, 2, 3 \qquad (20)$$

Where, $\Delta f_k$ is the sum of improvements in the values of the objective function obtained from the use of $k$ mutation strategy during $nt$ repetition. $\Delta Fes_k$ also shows the total number of the objective function evaluations for mutant solutions that have used $k$ mutation strategy during $nt$ repetition.

The pseudocode of the proposed approach is also interpreted as follows: first, the initial population is generated. Then, the entire population is divided into sub-populations. Each sub-population selects mutation and crossover operations independently and controls the setting of the parameters of $F$ and $CR$ based on adaptive and self-adaptive schemes of the parameters. Finally, when the termination conditions are provided, the best member of the population is presented as an optimum solution.

## 3. Experimental results

In this section, first, CIMPEDE approach performance is examined on 28 benchmark functions of CEC 2013 benchmark functions suite (Liang et al., 2013). These functions have some features, including unimodal, multi-modal, separable, non-separable, and composed of several benchmark functions. In addition, in their more objective function, rotation or change of global optimum position is applied, which makes achieving global optimization for any evolutionary algorithm extremely complicated, difficult and sometimes impossible. In order to conduct a competitive evaluation, the proposed approach CIMPEDE is compared with MPEDE algorithm and all comparable algorithms are compared with MPEDE in (Wu et al., 2016). These algorithms are JADE (Zhang & Sanderson, 2009), jDE (Brest et al., 2006), SaDE (Qin et al., 2008), EPSDE (Mallipeddi et al., 2011), and CoDE (Wang et al., 2011). The reason for choosing these algorithms is that CIMPEDE is an improved version of MPEDE in (Wu et al., 2016).

All experiments are in the state of minimization of benchmark functions on dimensions of 30 and 50. The results of each algorithm are based on two concepts of the mean and standard deviation of the solution error (obtained from the algorithm) relative to the actual solution of each benchmark function reported after 30 independent runs. In each Table, the best value for each benchmark function is displayed as bold. Also, a summary of Wilcoxon and Friedman statistical tests are presented at the end of each Table.

In addition, in order to evaluate CIMPEDE approach

power to solve real-world optimization problems, problem of optimizing an irrigation water allocation on Golestan Dam with the aim of decreasing irrigation deficiencies is examined. For competitive comparison, the proposed approach CIMPEDE is compared with JADE, jDE, SaDE, EPSDE, CoDE and MPEDE algorithms. All experiments conducted to solve this problem are also presented to evaluate of "minimizing irrigation deficiencies", or "minimizing absolute error between demand of different applicant sectors and released water". In solving this problem, all the results are reported in the five concepts of best, worst, mean, median and standard deviation from the absolute error between demand and released water. Also, the average of runtime (CPU time) between the comparable algorithms is evaluated and the Friedman test is used for a more comprehensive analysis.

### 3.1. Parameters settings

In order to provide a comprehensive comparison for CEC 2013 benchmark functions suite, each comparable algorithm 30 independent runs for implementations each benchmark function on two dimensions D = 30 and D = 50. In addition, the evolutionary loop of each algorithm repeats to $10000 \times \mathrm{D}$ times of the objective function evaluation. The setting for the parameters of the comparable algorithms is also based on the recommended values in their original research plan, and the specific parameters of CIMPEDE approach are set as follows: $NP = 240$, $nt = 20$ and $np_{elitist} = \sum_{k=1}^{3} np_{elitist_k}$, which is $np_{elitist_k} = 0.02 \times np_k$.

In order to solve the problem of optimizing the water allocation of Golestan Dam, the iteration number of evolutionary loop and population size ($NP$) for each algorithm are estimated at 8,000 and 240, respectively. Also, the number of objective function evaluations is considered in size $8000 \times NP$. Setting other parameters of the proposed approach is the same as the above.

### 3.3 Statistical tests

In order to illustrate the difference between the algorithms, statistical rank-based Wilcoxon test (Wilcoxon, 1992) is used with a significant level of $\alpha = 0.05$. The results of this test are expressed in three terms +, = and -, as follows:

- (+): The proposed algorithm provides better performance than another algorithm.
- (=): No significant difference is found between the two algorithms and equality evaluation has been recorded.
- (-): The proposed algorithm has a weaker performance than another algorithm.

Also, in order to rank the algorithms in terms of better performance, the average rank value (ARV) is performed using Friedman test (Derrac et al., 2011).

### 3.2. The experimental results and comparison of algorithms

The numerical results for the two states of mean and standard deviation of error values and statistical results of Wilcoxon and Friedman tests on CEC 2013 benchmark functions on each dimension of 30 and 50 are reported in Tables (1) and (2), respectively.

The results on the dimension 30, according to Table (1), shows that CIMPEDE approach is superior to MPEDE algorithm in F8, F9, F15, F16, F18, F20, F23, F24, F25, and F26 benchmark functions and has gained equal performance with MPEDE in F1-F7, F10, F12, F13, F17, F19, F21 and F27 benchmark functions. Also, CIMPEDE approach provides better performance than JADE, jDE, SaDE, EPSDE and CoDE algorithms in 20, 19, 20, 17, and 20 benchmark functions, respectively. In addition, the proposed algorithm shows equal performance for each of these algorithms in 6, 5, 4, 9, and 8 benchmark functions, respectively, and lower performance in 2, 4, 4, 2, and 0 benchmark functions, respectively.

According to Table (2) and at dimension 50, the CIMPEDE approach is noticeably competing with MPEDE algorithm. So that CIMPEDE approach in F4, F6, F8, F9, F13, F15-F18, F20, and F23-F27 benchmark functions had better performance, in F1, F2, F5, F7, F10, F12, F19, F21 and F28 benchmark functions had equal performance and in F3, F11, F14 and F22 benchmark functions had weaker performance than MPEDE. Also, CIMPEDE approach provides better performance than JADE, jDE, SaDE, EPSDE and CoDE algorithms in 20, 21, 20, 20, and 22 benchmark functions, respectively.In general, CIMPEDE in competition with other

comparable algorithms, had acceptable performance even with increasing dimensions of the benchmark functions. This algorithm is an effective approach on solving various complex benchmark functions with different features.

**Table 1.** Statistical-numerical results on 28 function of CEC 2013 benchmark functions suite on dimension 30

| F | | *JADE* | W | *jDE* | W | *SaDE* | W | *EPSDE* | W | *CoDE* | W | *MPEDE* | W | *CIMPEDE* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | Mean | 0.00E+00 | = | 0.00E+00 | = | 0.00E+00 | = | 0.00E+00 | = | 0.00E+00 | = | 0.00E+00 | = | 0.00E+00 |
| | STD | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 |
| F2 | Mean | 2.33E+04 | + | 9.15E+04 | + | 1.99E+04 | + | 2.21E+05 | + | 2.41E+05 | + | 1.34E+00 | = | 1.50E+00 |
| | STD | 9.71E+03 | | 4.56E+04 | | 1.25E+04 | | 1.22E+05 | | 1.75E+05 | | 2.86E+00 | | 2.03E+00 |
| F3 | Mean | 2.78E+05 | + | 3.15E+05 | + | 3.90E+06 | + | 1.38E+00 | = | 3.48E+05 | + | 1.15E+03 | = | 6.06E-02 |
| | STD | 5.62E+05 | | 6.06E+05 | | 4.13E+06 | | 3.09E+00 | | 7.38E+05 | | 2.06E+03 | | 9.03E-02 |
| F4 | Mean | 2.09E+00 | + | 4.35E+00 | + | 3.37E+02 | + | 5.28E+02 | + | 3.04E+01 | + | 2.91E-06 | = | 2.42E-09 |
| | STD | 4.25E+00 | | 3.86E+00 | | 1.85E+02 | | 1.32E+02 | | 1.10E+01 | | 3.97E-06 | | 2.94E-09 |
| F5 | Mean | 1.14E-13 | = | 1.36E-13 | = | 0.00E+00 | - | 0.00E+00 | - | 4.55E-14 | = | 1.14E-13 | = | 9.09E-14 |
| | STD | 0.00E+00 | | 5.08E-14 | | 0.00E+00 | | 0.00E+00 | | 6.23E-14 | | 0.00E+00 | | 5.08E-14 |
| F6 | Mean | 5.57E+00 | = | 7.39E+00 | + | 6.91E+00 | + | 1.13E+01 | = | 1.13E+01 | + | 1.75E-04 | = | 9.50E-01 |
| | STD | 1.16E+01 | | 4.23E+00 | | 4.46E+00 | | 1.38E+01 | | 9.22E+00 | | 3.91E-04 | | 2.12E+00 |
| F7 | Mean | 2.38E+00 | + | 6.19E+00 | + | 1.38E+01 | + | 4.55E+00 | + | 4.78E+01 | + | 6.57E+00 | = | 4.47E-01 |
| | STD | 1.08E+00 | | 2.48E+00 | | 1.01E+01 | | 2.45E+00 | | 3.62E+01 | | 4.98E+00 | | 3.66E-01 |
| F8 | Mean | 2.10E+01 | + | 2.09E+01 | + | 2.10E+01 | + | 2.09E+01 | + | 2.08E+01 | + | 2.09E+01 | + | 2.00E+01 |
| | STD | 4.57E-02 | | 6.11E-02 | | 4.69E-02 | | 3.96E-02 | | 1.41E-01 | | 2.99E-02 | | 1.93E-01 |
| F9 | Mean | 2.80E+01 | + | 2.14E+01 | + | 1.68E+01 | + | 3.06E+01 | + | 1.37E+01 | + | 1.42E+01 | + | 3.04E+00 |
| | STD | 1.57E+00 | | 5.77E+00 | | 2.62E+00 | | 2.31E+00 | | 2.75E+00 | | 5.31E+00 | | 2.20E-01 |
| F10 | Mean | 5.27E-02 | + | 4.63E-02 | + | 2.30E-01 | + | 3.94E-02 | = | 3.01E-02 | = | 1.43E-02 | = | 1.28E-02 |
| | STD | 3.70E-02 | | 1.25E-02 | | 1.26E-01 | | 2.79E-02 | | 1.35E-02 | | 9.60E-03 | | 1.01E-02 |
| F11 | Mean | 4.08E-11 | + | 0.00E+00 | - | 3.98E-01 | = | 0.00E+00 | - | 5.41E-09 | + | 2.27E-14 | - | 7.29E-12 |
| | STD | 2.46E-11 | | 0.00E+00 | | 5.45E-01 | | 0.00E+00 | | 7.94E-09 | | 3.11E-14 | | 1.22E-11 |
| F12 | Mean | 3.90E+01 | + | 4.15E+01 | + | 4.44E+01 | + | 7.33E+01 | + | 4.18E+01 | + | 2.39E+01 | = | 2.05E+01 |
| | STD | 4.80E+00 | | 9.60E+00 | | 5.39E+00 | | 4.69E+00 | | 1.37E+01 | | 5.40E+00 | | 5.20E+00 |
| F13 | Mean | 5.78E+01 | + | 7.05E+01 | + | 7.92E+01 | + | 1.02E+02 | + | 9.65E+01 | + | 4.96E+01 | = | 3.02E+01 |
| | STD | 1.27E+01 | | 1.41E+01 | | 2.84E+01 | | 2.54E+01 | | 2.24E+01 | | 2.49E+01 | | 1.09E+01 |
| F14 | Mean | 8.12E+00 | - | 3.33E-02 | - | 5.64E-01 | - | 1.23E+02 | = | 1.97E+02 | = | 8.74E+00 | - | 1.38E+02 |
| | STD | 1.94E+00 | | 3.16E-02 | | 9.93E-01 | | 3.13E+01 | | 4.38E+01 | | 2.55E+00 | | 5.56E+01 |
| F15 | Mean | 4.13E+03 | + | 3.86E+03 | + | 3.82E+03 | + | 5.71E+03 | + | 3.76E+03 | + | 4.18E+03 | + | 4.79E+02 |
| | STD | 5.07E+02 | | 7.80E+02 | | 1.48E+03 | | 2.33E+02 | | 4.75E+02 | | 6.39E+02 | | 8.74E+01 |
| F16 | Mean | 2.57E+00 | + | 2.35E+00 | + | 2.35E+00 | + | 2.11E+00 | + | 2.09E+00 | = | 2.53E+00 | + | 1.11E+00 |
| | STD | 7.76E-01 | | 3.07E-01 | | 1.12E-01 | | 3.54E-01 | | 7.48E-01 | | 3.40E-01 | | 1.59E-01 |
| F17 | Mean | 3.04E+01 | = | 3.04E+01 | = | 3.05E+01 | = | 3.04E+01 | = | 3.66E+01 | + | 3.06E+01 | = | 3.14E+01 |
| | STD | 2.79E-03 | | 1.76E-03 | | 4.35E-02 | | 4.04E-03 | | 1.70E+00 | | 5.68E-02 | | 3.58E+00 |
| F18 | Mean | 1.02E+02 | + | 1.34E+02 | + | 1.41E+02 | + | 1.58E+02 | + | 7.76E+01 | + | 9.34E+01 | + | 3.52E+01 |
| | STD | 8.75E+00 | | 1.76E+01 | | 1.36E+01 | | 8.88E+00 | | 2.06E+01 | | 1.11E+01 | | 6.24E+00 |
| F19 | Mean | 2.25E+00 | + | 1.29E+00 | - | 3.12E+00 | + | 2.35E+00 | + | 3.84E+00 | + | 1.83E+00 | = | 1.75E+00 |
| | STD | 1.69E-01 | | 1.19E-01 | | 5.89E-01 | | 1.74E-01 | | 1.13E+00 | | 3.52E-01 | | 2.69E-01 |
| F20 | Mean | 1.08E+01 | + | 1.10E+01 | + | 1.08E+01 | + | 1.17E+01 | + | 1.04E+01 | + | 1.08E+01 | + | 1.73E+00 |
| | STD | 6.06E-01 | | 7.22E-01 | | 5.47E-01 | | 3.73E-01 | | 3.19E-01 | | 6.61E-01 | | 1.36E-01 |
| F21 | Mean | 2.60E+02 | = | 3.57E+02 | = | 3.29E+02 | = | 3.09E+02 | = | 3.66E+02 | = | 3.29E+02 | = | 2.60E+02 |
| | STD | 5.48E+01 | | 7.86E+01 | | 6.42E+01 | | 8.69E+01 | | 1.12E+02 | | 6.42E+01 | | 5.48E+01 |
| F22 | Mean | 1.45E+02 | = | 9.27E+01 | - | 1.14E+02 | - | 3.60E+02 | = | 3.78E+02 | = | 1.33E+02 | - | 2.79E+02 |
| | STD | 5.44E+01 | | 4.39E+01 | | 2.52E+00 | | 8.43E+01 | | 1.17E+02 | | 2.92E+01 | | 1.22E+02 |
| F23 | Mean | 4.49E+03 | + | 4.14E+03 | + | 4.95E+03 | + | 5.57E+03 | + | 3.53E+03 | + | 4.33E+03 | + | 9.06E+02 |
| | STD | 3.69E+02 | | 3.82E+02 | | 1.29E+03 | | 4.58E+02 | | 8.39E+02 | | 5.07E+02 | | 2.33E+01 |
| F24 | Mean | 2.06E+02 | + | 2.13E+02 | + | 2.23E+02 | + | 2.68E+02 | + | 2.31E+02 | + | 2.10E+02 | + | 1.54E+02 |
| | STD | 6.36E+00 | | 8.97E+00 | | 8.70E+00 | | 8.56E+00 | | 1.08E+01 | | 3.09E+00 | | 1.01E+01 |
| F25 | Mean | 2.80E+02 | + | 2.59E+02 | + | 2.62E+02 | + | 2.74E+02 | + | 2.91E+02 | + | 2.94E+02 | + | 1.46E+02 |
| | STD | 8.17E+00 | | 1.07E+01 | | 5.11E+00 | | 1.78E+01 | | 1.82E+01 | | 1.11E+01 | | 1.18E+01 |
| F26 | Mean | 2.00E+02 | + | 2.00E+02 | + | 2.00E+02 | + | 2.00E+02 | + | 2.00E+02 | + | 2.00E+02 | + | 1.42E+02 |
| | STD | 2.06E-03 | | 1.67E-03 | | 1.16E-03 | | 7.54E-03 | | 4.90E-03 | | 8.94E-05 | | 6.04E+00 |
| F27 | Mean | 7.36E+02 | + | 5.86E+02 | + | 5.50E+02 | + | 1.07E+03 | + | 5.72E+02 | + | 3.69E+02 | = | 3.78E+02 |
| | STD | 8.99E+01 | | 1.61E+02 | | 5.46E+01 | | 3.76E+01 | | 9.31E+01 | | 2.04E+01 | | 6.91E+00 |
| F28 | Mean | 3.00E+02 | - | 3.00E+02 | = | 3.00E+02 | - | 3.00E+02 | = | 3.00E+02 | = | 3.00E+02 | - | 3.00E+02 |
| | STD | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 |
| +, ≅, − | | 20/6/2 | | 19/5/4 | | 20/4/4 | | 17/9/2 | | 20/8/0 | | 10/14/4 | | - |
| | ARV | 4.21 | | 4.09 | | 4.43 | | 5.07 | | 4.86 | | 3.27 | | 2.10 |

**Table 2.** Statistical-numerical results on 28 functions of CEC 2013 benchmark functions suite on dimension 50

| F | | *JADE* | W | *jDE* | W | *SaDE* | W | *EPSDE* | W | *CoDE* | W | *MPEDE* | W | *CIMPEDE* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | Mean | 2.27E-13 | = | 1.82E-13 | = | 0.00E+00 | = | 0.00E+00 | = | 0.00E+00 | = | 1.82E-13 | = | 1.36E-13 |
| | STD | 0.00E+00 | | 1.02E-13 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 1.02E-13 | | 1.25E-13 |
| F2 | Mean | 1.00E+05 | = | 3.34E+05 | + | 1.57E+05 | = | 2.07E+06 | + | 6.06E+05 | + | 9.70E+04 | = | 1.01E+05 |
| | STD | 5.36E+04 | | 1.04E+05 | | 5.19E+04 | | 9.78E+05 | | 2.08E+05 | | 5.88E+04 | | 5.18E+04 |
| F3 | Mean | 1.57E+07 | = | 6.65E+06 | + | 1.47E+07 | + | 1.96E+05 | = | 2.91E+07 | + | 2.24E+04 | - | 1.07E+05 |
| | STD | 9.28E+06 | | 9.00E+06 | | 1.56E+07 | | 1.91E+05 | | 1.64E+07 | | 4.69E+04 | | 7.17E+04 |
| F4 | Mean | 9.58E+03 | + | 3.07E+01 | + | 1.31E+03 | + | 8.77E+03 | + | 1.65E+02 | + | 2.96E+00 | + | 3.60E-03 |
| | STD | 2.14E+04 | | 1.85E+01 | | 7.93E+02 | | 1.28E+03 | | 9.07E+01 | | 5.68E+00 | | 5.58E-03 |
| F5 | Mean | 2.05E-13 | + | 8.41E-13 | + | 0.00E+00 | - | 9.09E-14 | = | 9.09E-14 | = | 1.14E-13 | = | 1.14E-13 |
| | STD | 5.08E-14 | | 6.05E-13 | | 0.00E+00 | | 5.08E-14 | | 5.08E-14 | | 0.00E+00 | | 0.00E+00 |
| F6 | Mean | 4.34E+01 | + | 4.34E+01 | + | 5.18E+01 | + | 4.34E+01 | + | 4.34E+01 | + | 4.34E+01 | + | 2.40E+01 |
| | STD | 0.00E+00 | | 2.37E-04 | | 1.88E+01 | | 0.00E+00 | | 1.70E-04 | | 0.00E+00 | | 1.08E+01 |
| F7 | Mean | 2.64E+01 | + | 2.79E+01 | + | 4.84E+01 | + | 6.97E+01 | + | 6.97E+01 | + | 1.57E+01 | = | 8.74E+00 |
| | STD | 9.67E+00 | | 7.19E+00 | | 1.48E+01 | | 1.63E+01 | | 3.75E+01 | | 6.67E+00 | | 5.31E+00 |
| F8 | Mean | 2.11E+01 | + | 2.11E+01 | + | 2.12E+01 | + | 2.11E+01 | + | 2.11E+01 | + | 2.11E+01 | + | 2.11E+01 |
| | STD | 7.64E-02 | | 5.13E-02 | | 4.18E-02 | | 4.37E-02 | | 8.58E-02 | | 3.30E-02 | | 2.64E-02 |
| F9 | Mean | 5.39E+01 | + | 4.08E+01 | + | 3.58E+01 | + | 5.77E+01 | + | 2.96E+01 | + | 3.09E+01 | + | 3.17E+00 |
| | STD | 8.91E-01 | | 1.32E+01 | | 7.20E+00 | | 3.13E+00 | | 5.23E+00 | | 2.89E+00 | | 4.04E-01 |
| F10 | Mean | 6.21E-02 | + | 1.91E-01 | + | 1.46E-01 | + | 9.56E-02 | + | 4.97E-02 | + | 1.33E-02 | = | 1.48E-02 |
| | STD | 4.06E-02 | | 1.04E-01 | | 8.41E-02 | | 3.27E-02 | | 9.10E-03 | | 1.21E-02 | | 7.98E-03 |
| F11 | Mean | 1.60E-11 | - | 4.55E-14 | - | 1.39E+00 | = | 0.00E+00 | - | 2.04E+00 | + | 5.68E-14 | - | 1.07E-06 |
| | STD | 1.31E-11 | | 2.54E-14 | | 1.13E+00 | | 0.00E+00 | | 1.62E+00 | | 0.00E+00 | | 1.65E-06 |
| F12 | Mean | 7.71E+01 | + | 9.12E+01 | + | 9.79E+01 | + | 1.81E+02 | + | 1.13E+02 | + | 4.74E+01 | = | 3.77E+01 |
| | STD | 6.60E+00 | | 1.31E+01 | | 1.08E+01 | | 2.52E+01 | | 1.77E+01 | | 9.51E+00 | | 6.48E+00 |
| F13 | Mean | 1.52E+02 | + | 1.80E+02 | + | 2.28E+02 | + | 2.51E+02 | + | 1.85E+02 | + | 1.55E+02 | + | 3.40E+01 |
| | STD | 2.70E+01 | | 2.34E+01 | | 3.24E+01 | | 3.24E+01 | | 5.65E+01 | | 1.96E+01 | | 4.99E+00 |
| F14 | Mean | 2.38E+01 | - | 1.89E-01 | - | 5.48E+00 | - | 1.28E+02 | - | 5.99E+02 | + | 5.07E+00 | - | 3.81E+02 |
| | STD | 3.73E+00 | | 3.19E-01 | | 4.08E+00 | | 3.89E+01 | | 1.91E+02 | | 2.31E+00 | | 9.62E+01 |
| F15 | Mean | 8.23E+03 | + | 9.02E+03 | + | 8.83E+03 | + | 1.09E+04 | + | 6.91E+03 | + | 8.81E+03 | + | 4.81E+02 |
| | STD | 5.57E+02 | | 6.08E+02 | | 2.50E+03 | | 6.31E+02 | | 5.33E+02 | | 3.66E+02 | | 8.36E+01 |
| F16 | Mean | 3.23E+00 | + | 3.32E+00 | + | 2.81E+00 | + | 2.55E+00 | + | 1.38E+00 | = | 3.36E+00 | + | 9.42E-01 |
| | STD | 2.66E-01 | | 3.29E-01 | | 4.11E-01 | | 2.66E-01 | | 6.58E-01 | | 1.69E-01 | | 1.51E-01 |
| F17 | Mean | 5.09E+01 | + | 5.08E+01 | + | 5.16E+01 | + | 5.09E+01 | + | 7.97E+01 | + | 5.09E+01 | + | 3.05E+01 |
| | STD | 4.08E-02 | | 4.87E-03 | | 7.39E-01 | | 1.32E-02 | | 2.78E+00 | | 7.39E-02 | | 1.04E+01 |
| F18 | Mean | 1.93E+02 | + | 2.18E+02 | + | 1.87E+02 | + | 2.96E+02 | + | 1.21E+02 | + | 1.49E+02 | + | 3.20E+01 |
| | STD | 1.66E+01 | | 3.03E+01 | | 7.53E+01 | | 8.62E+00 | | 2.24E+01 | | 3.73E+01 | | 7.89E+00 |
| F19 | Mean | 4.29E+00 | + | 2.17E+00 | - | 1.08E+01 | + | 4.09E+00 | + | 7.45E+00 | + | 3.22E+00 | = | 3.12E+00 |
| | STD | 2.12E-01 | | 2.36E-01 | | 2.86E+00 | | 2.58E-01 | | 2.14E+00 | | 1.41E-01 | | 4.24E-01 |
| F20 | Mean | 1.98E+01 | + | 2.10E+01 | + | 1.91E+01 | + | 2.15E+01 | + | 1.98E+01 | + | 1.90E+01 | + | 1.78E+00 |
| | STD | 5.56E-01 | | 5.45E-01 | | 1.23E+00 | | 5.30E-01 | | 7.39E-01 | | 6.95E-01 | | 1.11E-01 |
| F21 | Mean | 9.38E+02 | = | 7.53E+02 | = | 7.53E+02 | = | 5.69E+02 | = | 7.53E+02 | = | 9.38E+02 | = | 2.79E+02 |
| | STD | 4.12E+02 | | 5.05E+02 | | 5.05E+02 | | 5.05E+02 | | 5.05E+02 | | 4.12E+02 | | 1.09E+02 |
| F22 | Mean | 6.30E+01 | - | 1.48E+01 | - | 2.45E+01 | - | 2.75E+02 | - | 8.85E+02 | = | 1.77E+01 | - | 6.90E+02 |
| | STD | 1.91E+01 | | 3.12E+00 | | 3.85E+00 | | 1.05E+02 | | 6.76E+02 | | 1.51E+00 | | 5.04E+01 |
| F23 | Mean | 8.50E+03 | + | 8.54E+03 | + | 9.30E+03 | + | 1.10E+04 | + | 7.61E+03 | + | 8.92E+03 | + | 8.09E+02 |
| | STD | 9.03E+02 | | 1.11E+03 | | 2.03E+03 | | 3.88E+02 | | 6.18E+02 | | 9.64E+02 | | 8.42E+01 |
| F24 | Mean | 2.49E+02 | + | 2.55E+02 | + | 2.68E+02 | + | 3.51E+02 | + | 2.60E+02 | + | 2.40E+02 | + | 1.45E+02 |
| | STD | 2.58E+01 | | 1.92E+01 | | 9.94E+00 | | 8.81E+00 | | 5.51E+00 | | 9.65E+00 | | 1.28E+01 |
| F25 | Mean | 3.51E+02 | + | 3.13E+02 | + | 3.45E+02 | + | 3.74E+02 | + | 3.71E+02 | + | 3.63E+02 | + | 1.43E+02 |
| | STD | 2.44E+01 | | 8.25E+00 | | 4.89E+00 | | 6.27E+00 | | 1.14E+01 | | 1.48E+01 | | 4.67E+00 |
| F26 | Mean | 2.50E+02 | + | 3.09E+02 | + | 3.09E+02 | + | 3.99E+02 | + | 3.03E+02 | + | 2.26E+02 | + | 1.37E+02 |
| | STD | 1.06E+02 | | 1.03E+02 | | 1.00E+02 | | 1.11E+02 | | 9.38E+01 | | 5.87E+01 | | 3.96E+00 |
| F27 | Mean | 1.47E+03 | + | 1.06E+03 | + | 1.16E+03 | + | 1.71E+03 | + | 1.12E+03 | + | 9.00E+02 | + | 3.75E+02 |
| | STD | 2.42E+02 | | 9.59E+01 | | 8.89E+01 | | 4.31E+01 | | 1.32E+02 | | 8.59E+01 | | 7.47E+00 |
| F28 | Mean | 4.00E+02 | = | 9.92E+02 | = | 4.00E+02 | - | 4.00E+02 | = | 4.00E+02 | = | 4.00E+02 | = | 3.95E+02 |
| | STD | 0.00E+00 | | 1.32E+03 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 0.00E+00 | | 1.09E+01 |
| +,≅,− | | 20/5/3 | | 21/3/4 | | 20/4/4 | | 20/5/3 | | 22/6/0 | | 15/9/4 | | - |
| | ARV | 4.16 | | 4.07 | | 4.50 | | 5.29 | | 4.59 | | 3.39 | | 2.00 |

### 3.4. The experimental results for irrigation water allocation optimization problem

In this section, the values of released water of Golestan Dam in proportion to the demand in the $t$-period (which is presented with a ten-year vision horizon in 1-month periods) are optimized to minimize irrigation deficiencies and ensure the maximization of downstream requirements in different applicant sectors. In order to evaluate and comprehensive analysis of the CIMPEDE proposed approach with other advanced algorithms, 5 different runs have been performed to solve the water allocation optimization problem. Numerical results after 5 different runs and the results analysis based on the Wilcoxon statistical test on irrigation water allocation problem is reported in Table (3). The best, worst, mean, median and standard deviation of absolute error values between demand and released water for each of the algorithms participating in the competition and the mean runtime of the algorithms

are shown in Table (4). In addition, the ranking of algorithms based on Friedman statistical test is presented at the bottom of Table (4).

Table (4) clearly shows that CIMPEDE is superior to JADE, jDE, SaDE, EPSDE, CoDE and MPEDE with a big difference. The best performance in providing a quality solution based on Table (4) with mean absolute error 0 MCM is assigned to CIMPEDE. This is while, the second best algorithm cannot get a mean error less than 1.81E-220 MCM. Nevertheless, CIMPEDE needs a long runtime than all other comparable algorithms. Because CIMPEDE is a combination of several algorithms and various techniques. Friedman statistical test also records the first rank for CIMPEDE and the next ranks are awarded to JADE, MPEDE, CoDE, jDE, EPSDE and SaDE, respectively. The convergence diagrams obtained from the best values during the evolutionary process for all of the algorithms participating in the competition are presented in Fig. 4. In addition, the diagrams of the mean irrigation deficiencies errors between released water values by CIMPEDE algorithm ($R_t$) and real demand ($D_t$) assuming each $t \in [1, 2, \dots, 120]$ and based on 5 runs are shown in Fig. 5. Fig. 5 proves the almost complete correlation of the water released values in comparison with the real demand for CIMPEDE.

In this study, several other diagrams have been drawn to measure the efficiency of algorithms in correctly allocating released water values in comparison with the total demand and also for find the number of algorithm failures in the incorrect allocation of released water in proportion to the total demand. For this purpose, in Fig. 6, the analysis of the two indicators of reliability and vulnerability in the safe state is expressed in terms of percentage. According to Fig. 6, the CIMPEDE algorithm achieves the best reliability with an optimum efficiency of 94% in during of 120-periodes among the comparable algorithms. However, the next highest performance with a mean reliability of 91% is assigned to JADE. On the other hand, based on the vulnerability index, the best performance belongs to the CIMPEDE. So that, CIMPEDE with the mean failure rate of 0%, receives the highest evaluation in recording the lowest number of failures.
The difference of mean error values between the total demand and released water in an annual average after 5 different runs is shown in Fig. 7. The bar diagrams shown in Fig. 7, are allocate the highest intensity of violations in the released water values to the SaDE and EPSDE algorithms, respectively. In this test, the lowest intensity of violation among all comparable algorithms is assigned to the CIMPEDE. CIMPEDE in between all years does not report intensity of violations in the released water values greater than 1E-18 MCM. Base on comparison of CIMPEDE with MPEDE, the efficiency of CIMPEDE is clearly evident. So that, MPEDE can never achieve a violation intensity of less than 1E-15 MCM. Thus, CIMPEDE is able to provide efficiency, stability, and accuracy in finding better solutions to solve water engineering problems, which can be widely used in the researches.

## 3. Conclusions

This study is presented in other to minimizing irrigation shortages by focusing on the optimal allocation of water resources. A case study on Golestan Dam is modeled in the form of single-reservoir system and is used from an improved differential evolution (DE) algorithm based on several techniques. Applying multi-population structures with an ensemble of mutation strategies and using adaptive and self-adaptive schemes during the evolutionary process can significantly improve the performance of DE. For this purpose, the proposed approach CIMPEDE has been introduced in this study using these three techniques. CIMPEDE provides high-quality solutions from combining several algorithms, an ensemble of efficient mutation strategies, adaptive and self-adaptive schemes of parameters, as well as the proper detection of each time they are used in each repeat of the evolutionary process. The analysis of the extensive experimental results on CEC 2013 benchmark functions suite and the irrigation water allocation optimization problem has shown that the type of mutation strategies and the setting of their related parameters, depending on the type of problem and even during the evolutionary process of a specific benchmark function, are completely different. This is clearly based on numerical-statistical results.

**Table 3.** Statistical-numerical results of irrigation water allocation optimization problem on Golestan Dam after 5 different runs

| Run | *JADE* | W | *jDE* | W | *SaDE* | W | *EPSDE* | W | *CoDE* | W | *MPEDE* | W | *CIMPEDE* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.54E-224 | + | 7.77E-29 | + | 2.93E-15 | + | 2.51E-25 | + | 2.43E-30 | + | 4.72E-288 | = | **0** |
| 2 | 3.87E-228 | + | 8.27E-27 | + | 8.30E-15 | + | 1.25E-25 | + | 4.45E-74 | + | 9.71E-30 | + | **0** |
| 3 | 9.03E-220 | + | 2.58E-27 | + | 3.94E-15 | + | 1.90E-25 | + | 1.20E-74 | + | 8.53E-283 | = | **0** |
| 4 | 1.33E-229 | + | 2.49E-27 | + | 8.99E-15 | + | 1.20E-24 | + | 3.88E-29 | + | 3.88E-29 | + | **0** |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 4.55E-231 | + | 6.63E-28 | + | 8.78E-15 | + | 8.36E-25 | + | 6.21E-28 | + | 8.92E-264 | = | **0** |
| +, ≅, − | 5/0/0 | | 5/0/0 | | 5/0/0 | | 5/0/0 | | 5/0/0 | | 2/3/0 | | - |

**Table 4.** Experimental results based on statistical analysis of reported error values in solving of irrigation water allocation optimization problem on Golestan Dam

| | ***JADE*** | ***jDE*** | ***SaDE*** | ***EPSDE*** | ***CoDE*** | ***MPEDE*** | ***CIMPEDE*** |
|---|---|---|---|---|---|---|---|
| **Best** | 4.55E-231 | 7.77E-29 | 2.93E-15 | 1.25E-25 | 1.20E-74 | 4.72e-288 | **0** |
| **Mean** | 1.81E-220 | 2.82E-27 | 6.59E-15 | 5.20E-25 | 1.33E-28 | 9.71e-30 | **0** |
| **Median** | 3.87E-228 | 2.49E-27 | 8.30E-15 | 2.51E-25 | 2.43E-30 | 8.92e-264 | **0** |
| **STD** | 0 | 3.24E-27 | 2.91E-15 | 4.73E-25 | 2.74E-28 | 1.68e-29 | **0** |
| **Worst** | 9.03E-220 | 8.27E-27 | 8.99E-15 | 1.20E-24 | 6.21E-28 | 3.88e-29 | **0** |
| **CPU Time (sec)** | 28.65 | 21.90 | 40.98 | **21.77** | 38.07 | 34.32 | 45.44 |
| **Friedman test ranking** | 2.60 | 5.00 | 7.00 | 6.00 | 3.70 | 2.70 | **1.00** |

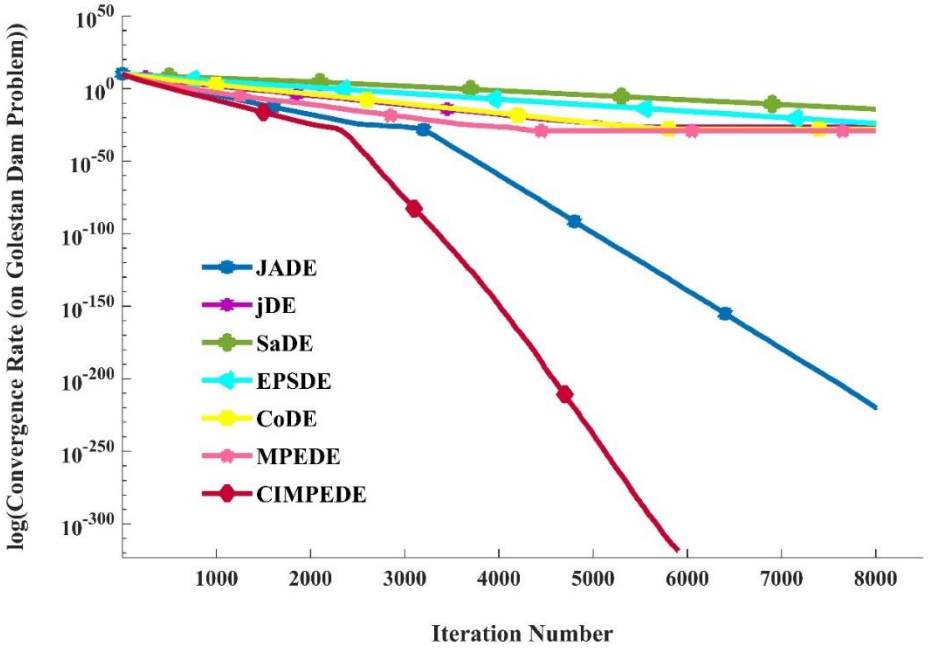



**Fig. 4.** Study of convergence rates between comparable algorithms in solving of irrigation water allocation problem

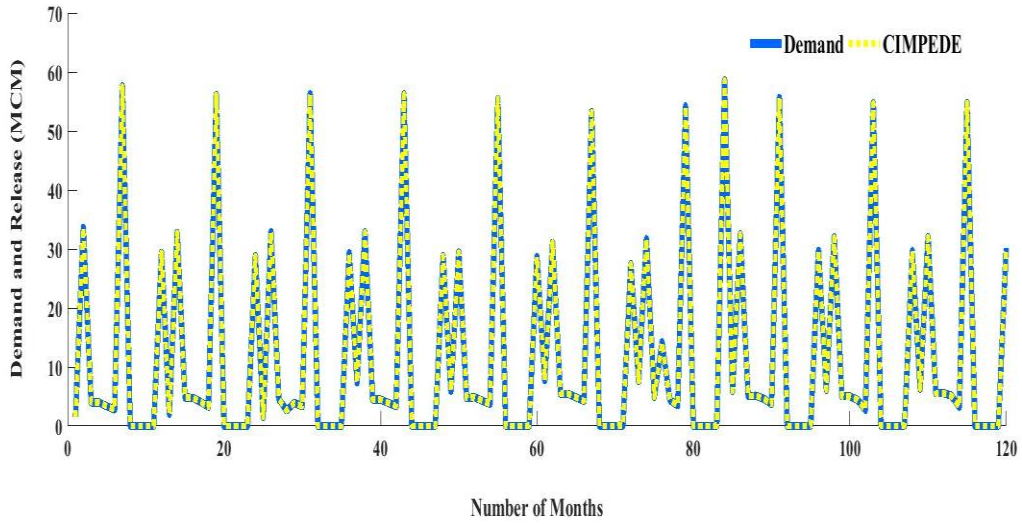



**Fig. 5.** Comparison of real demand values and released water values of Golestan Dam by CIMPEDE algorithm in a 10-year period

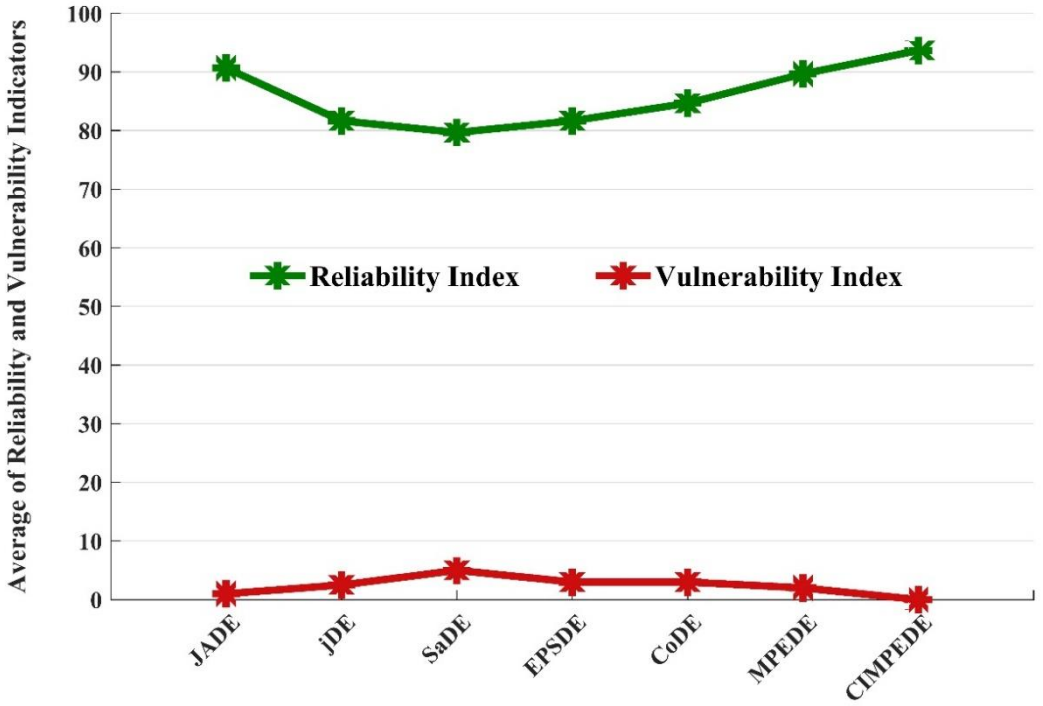



**Fig. 6.** Comparison of mean reliability and failure values between comparable algorithms based on reliability index and vulnerability index

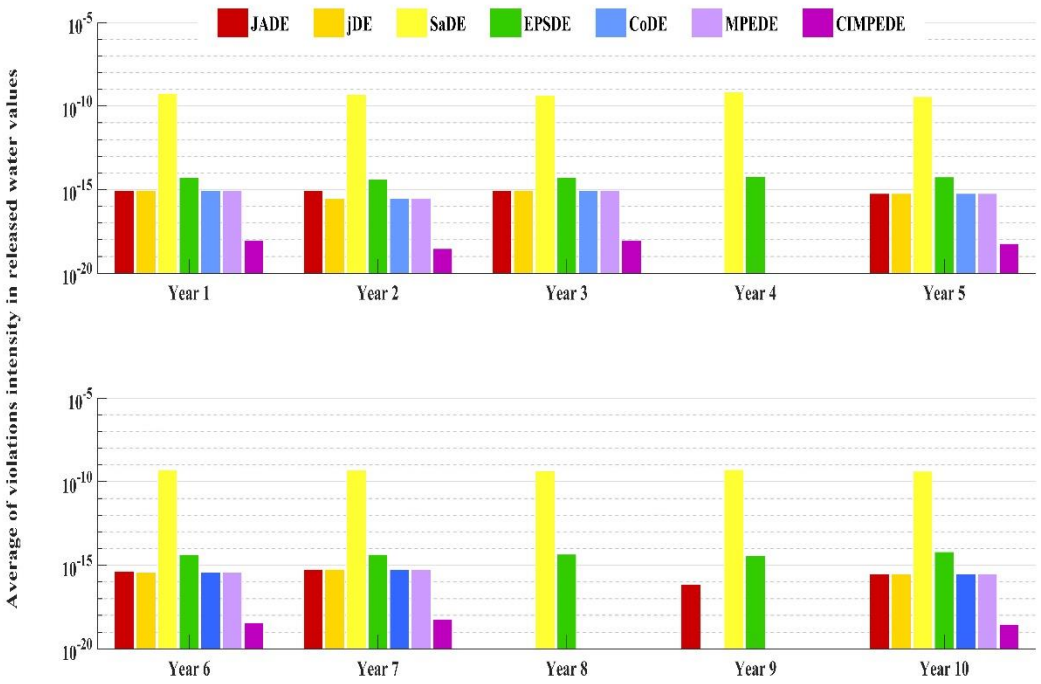



**Fig. 7.** Comparison of mean error values between total demand and released water in an annual average (in 10-years period) between comparable algorithms

## 4. Acknowledgement

The author acknowledge with grateful appreciation the information assistance and data set support provided by the Golestan Regional Water Department.

## 5. Reference

1. Ali, M. Z., Awad, N. H., & Suganthan, P. N. (2015). "Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization." Applied Soft Computing, 33, 304-327.
2. Bazaraa, M. S., Jarvis, J. J., & Sherali, H. D. (2008). Linear programming and network flows, John Wiley & Sons.
3. Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). "A survey on optimization metaheuristics." Information sciences, 237, 82-117.
4. Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems." IEEE transactions on evolutionary computation, 10(6), 646-657.
5. Cheng, J., Pan, Z., Liang, H., Gao, Z., & Gao, J. (2021). "Differential evolution algorithm with fitness and diversity ranking-based mutation operator." Swarm and Evolutionary Computation, 61, 100816.
6. Cui, L., Li, G., Lin, Q., Chen, J., & Lu, N. (2016). "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations." Computers & Operations Research, 67, 155-173.
7. Deng, W., Xu, J., Song, Y., & Zhao, H. (2021). "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem." Applied Soft Computing, 100, 106724.
8. Derrac, J., García, S., Molina, D., & Herrera, F. (2011). "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms." Swarm and Evolutionary Computation, 1(1), 3-18.
9. Dorigo, M., & Di Caro, G. "Ant colony optimization: a new meta-heuristic." Proceedings of the 1999 congress on evolutionary

computation-CEC99 (Cat. No. 99TH8406), 1470-1477.
10. Draa, A., Bouzoubia, S., & Boukhalfa, I. (2015). "A sinusoidal differential evolution algorithm for numerical optimisation." Applied Soft Computing, 27, 99-126.
11. Duan, M., Yang, H., Wang, S., & Liu, Y. (2019). "Self-adaptive dual-strategy differential evolution algorithm." Plos one, 14(10), e0222706.
12. Eberhart, R., & Kennedy, J. "A new optimizer using particle swarm theory." MHS'95. Proceedings of the sixth international symposium on micro machine and human science, 39-43.
13. Ehteram, M., Karami, H., & Farzin, S. (2018). "Reducing irrigation deficiencies based optimizing model for multi-reservoir systems utilizing spider monkey algorithm." Water Resources Management, 32(7), 2315-2334.
14. Fan, Q., Wang, W., & Yan, X. (2019). "Differential evolution algorithm with strategy adaptation and knowledge-based control parameters." Artificial Intelligence Review, 51(2), 219-253.
15. Holland, J. H. (1992). "Genetic algorithms." Scientific american, 267(1), 66-73.
16. Islam, S. M., Das, S., Ghosh, S., Roy, S., & Suganthan, P. N. (2011). "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(2), 482-500.
17. Li, X., Wang, L., Jiang, Q., & Li, N. (2021). "Differential evolution algorithm with multi-population cooperation and multi-strategy integration." Neurocomputing, 421, 285-302.
18. Li, Y., Wang, S., & Yang, B. (2020). "An improved differential evolution algorithm with dual mutation strategies collaboration." Expert Systems with Applications, 153, 113451.
19. Liang, J. J., Qu, B., Suganthan, P. N., & Hernández-Díaz, A. G. (2013). "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization." Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, 201212(34), 281-295.
20. Ma, Y., & Bai, Y. (2020). "A multi-population differential evolution with best-random mutation strategy for large-scale global optimization." Applied Intelligence, 50(5), 1510-1526.
21. Mallipeddi, R., Suganthan, P. N., Pan, Q.-K., & Tasgetiren, M. F. (2011). "Differential evolution algorithm with ensemble of parameters and mutation strategies." Applied soft computing, 11(2), 1679-1696.
22. Qin, A. K., Huang, V. L., & Suganthan, P. N. (2008). "Differential evolution algorithm with strategy adaptation for global numerical optimization." IEEE transactions on Evolutionary Computation, 13(2), 398-417.
23. Ruigar, H., & Golian, S. (2016). "Prediction of precipitation in Golestan dam watershed using climate signals." Theoretical and applied climatology, 123(3), 671-682.
24. Saadat, H., Adamowski, J., Bonnell, R., Sharifi, F., Namdar, M., & Ale-Ebrahim, S. (2011). "Land use and land cover classification over a large area in Iran based on single date analysis of satellite imagery." ISPRS journal of Photogrammetry and Remote Sensing, 66(5), 608-619.
25. Salehinejad, H., Rahnamayan, S., & Tizhoosh, H. R. (2017). "Micro-differential evolution: Diversity enhancement and a comparative study." Applied Soft Computing, 52, 812-833.
26. Sharifi, F., Samadi, S. Z., & Wilson, C. A. (2012). "Causes and consequences of recent floods in the Golestan catchments and Caspian Sea regions of Iran." Natural hazards, 61(2), 533-550.
27. Song, E., & Li, H. (2021). "A self-adaptive differential evolution algorithm using oppositional solutions and elitist sharing." IEEE Access, 9, 20035-20050.
28. Storn, R. "On the usage of differential evolution for function optimization." Proceedings of North American Fuzzy Information Processing, 519-523.
29. Storn, R., & Price, K. (1997). "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." Journal of global optimization, 11(4), 341-359.
30. Wang, Y., Cai, Z., & Zhang, Q. (2011). "Differential evolution with composite trial vector generation strategies and control parameters." IEEE transactions on evolutionary computation, 15(1), 55-66.
31. Wilcoxon, F. (1992). "Individual comparisons by ranking methods." Breakthroughs in statistics, Springer, 196-202.
32. Wu, G., Mallipeddi, R., Suganthan, P. N., Wang, R., & Chen, H. (2016). "Differential evolution with multi-population based ensemble of mutation strategies." Information Sciences, 329, 329-345.
33. Xiang, W.-l., Meng, X.-l., An, M.-q., Li, Y.-z., & Gao, M.-x. (2015). "An enhanced differential evolution algorithm based on multiple mutation

strategies." Computational intelligence and neuroscience, 2015.
34. Xu, B., Tao, L., Chen, X., & Cheng, W. (2019). "Adaptive differential evolution with multi-population-based mutation operators for constrained optimization." Soft Computing, 23(10), 3423-3447.
35. Zhang, J., & Sanderson, A. C. (2009). "JADE: adaptive differential evolution with optional external archive." IEEE Transactions on evolutionary computation, 13(5), 945-958.
36. Zhang, Q., Liu, W., Meng, X., Yang, B., & Vasilakos, A. V. (2017). "Vector coevolving particle swarm optimization algorithm." Information sciences, 394, 273-298.
37. Zhou, Z., Abawajy, J., Shojafar, M., & Chowdhury, M. (2022). "DEHM: An Improved Differential Evolution Algorithm using Hierarchical Multi-strategy in a Cybertwin 6G Network." IEEE Transactions on Industrial Informatics.